Admissible generalizations of examples as rules (2019)

Philippe Besnard, Thomas Guyet, Véronique Masson

Admissible Generalizations of Examples as Rules

Philippe Besnard
CNRS – IRIT
France

Thomas Guyet
Agrocampus Ouest – IRISA
France

Véronique Masson
Université de Rennes 1 – IRISA
France

***Abstract*— Rule learning is a data analysis task that consists in extracting rules that generalize examples. This is achieved by a plethora of algorithms. Some generalizations make more sense for the data scientists, called here admissible generalizations. The purpose of this article is to show formal properties of admissible generalizations. A formalization for generalization of examples is proposed allowing the expression of rule admissibility. Some admissible generalizations are captured by preclosure and capping operators. Also, we are interested in selecting supersets of examples that induce such operators. We then define classes of selection functions. This formalization is more particularly developed for examples with numerical attributes. Classes of such functions are associated with notions of generalization and they are used to comment some results of the CN2 algorithm [5].**

I. Introduction

Generalizing a given set of examples is essential in many machine learning techniques such as rule learning or pattern mining. Particularly, rule learning is a data mining task that consists in generating disjunctive sets of rules from a dataset of examples labeled by a class identifier. We are focusing on propositional rule induction [8] where rules have the form "`IF` *Conditions* `THEN` *class-label*". Rule learning consists in finding individual rules [19], each rule generalizes a subset of the dataset examples. Many algorithms achieve rule induction, from CN2 [5], Ripper [6] to recent subgroup discovery algorithms [3]. Usually each rule is evaluated by different measures using the number of positive and negative examples covered, i.e. generalized, by the rule. Numerous interestingness measures [11] on rules have been proposed. Some heuristic measures guide the machine learning algorithms and some of them are used in a post-processing step to select final rules.

Rule learning algorithms received recent attention in the machine learning community due to their *interpretability*. Explainability and interpretability of machine learning results is a hot topic [7]. The logical structure of a rule can be easily interpreted by users not familiar with machine learning or data mining concepts. We feel that generalization of examples by machine learning algorithms impacts interpretability, particularly when this generalization is counter-intuitive.

Table I is an illustration of a dataset of house rental ads. Each row is a house rental ad and each column is an attribute. With a minimal coverage size of 2, CN2 extracts the following three rules predicting a value for the class-attribute C:

- $\pi_1^{CN2} : A_5 = \text{Downtown} \rightarrow C = \text{expensive}$
- $\pi_2^{CN2} : A_2 < 2.50 \wedge A_4 = \text{Toulouse} \rightarrow C = \text{low-priced}$
- $\pi_3^{CN2} : A_1 > 36.00 \wedge A_3 = D \rightarrow C = \text{cheap}$

§ Corresponding author: Véronique Masson (veronique.masson@irisa.fr)

TABLE I
Dataset of house rental ads. Blank cells are missing values.

	(C) *Price*	(A_1) *Area*	(A_2) *#Rooms*	(A_3) *Energy*	(A_4) *Town*	(A_5) *District*	(A_6) *Exposure*
1	low-priced	45	2	D	Toulouse	Minimes	
2	cheap	75	4	D	Toulouse	Rangueil	
3	expensive	65	3		Toulouse	Downtown	
4	low-priced	32	2	D	Toulouse		SE
5	mid-priced	65	2	D	Rennes		SW
6	expensive	100	5	C	Rennes	Downtown	
7	cheap	40	2	D	Betton		S

Generalization of a metric attribute leads to the difficult question of defining boundary values. The π_2^{CN2} rule uses a value for A_2 attribute (i.e., 2.5) which is not in the original dataset. The choice of this boundary is motivated by statistical reasons: with an hypothesis of uniform distribution of numerical attribute, it minimizes the generalization error. One can notice that it is less intuitive (although equivalent) than the rule: $A_2 \leq \mathbf{2} \wedge A_4 = \text{Toulouse} \rightarrow C = \text{low-priced}$.

This small example illustrates that existing rule learning algorithms underestimate the effects that their underlying hypotheses about generalization can have on the value of extracted rules for a data scientist – some rules sometimes fail to capture an intuitive generalization of the examples.

We are wondering whether it is possible to highlight some general principles of intuitive generalization that would help to analyze or to qualify rules or rulesets extracted by rule learning algorithms. This means that we are interested in analyzing consequences of choices made by rule learning algorithm when generalizing examples.

There are different approaches to reach such an objective: scoring interestingness or quality measures [1] or analyzing results on the light of subjective criteria [20] (see related works for more details).

The purpose of this paper is to propose a topological formalization for generalization of examples which favours an analysis on the *admissibility* of the generalization by enabling to express different notions of admissibility. One objective is to make it possible to compare the outputs of rule learning algorithms with the theoretically admissible rules in order to shed light on some poorly interpretable outputs.

Importantly, our work is also original as it pays special attention to the values (mostly as boundaries) occurring in rules whereas work in the literature on improving rules or rulesets usually focus on the structure of rules or of rulesets, see e.g. [13], [4].

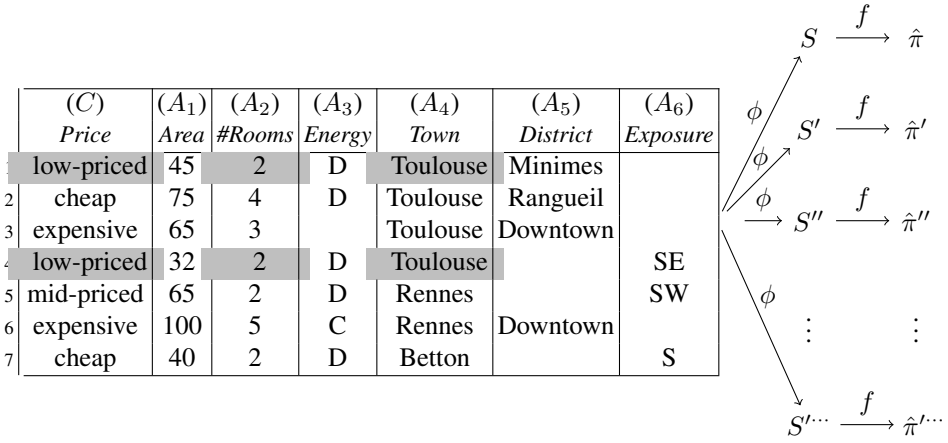

Fig. 1. Abstract modeling of the LearnOneRule process (see text for details). Grey cells illustrate selected rows and columns that may generate π_2^{CN2}.

The contributions of this work are:

- we propose an abstract formalization of the rule learning process introducing the generalization of examples as a choice of a generalization rule $\pi = \widehat{S}$ (for a set of examples S) among supersets of S,
- we introduce the notion of admissibility, we derive two alternative versions of admissibility from Kuratowski's axioms and we give sufficient conditions on choice functions to induce admissible generalizations,
- we instantiate admissibility in the specific case of metric attributes and we analyze some CN2 results in the light of our framework.

Note that we do not have an immediate practical objective: our purpose is not to design a new rule learning algorithm but to set a general framework that may help to shed light on some aspects of existing, or future, rule learning algorithms. In particular, we illustrate that a well-known algorithm such as CN2 makes some counterintuitive choices of boundaries in rules with metric attributes.

II. Generalization of data as a rule

This work focuses on the LearnOneRule step of the rule learning process [19], [8]. The LearnOneRule process is viewed as a two-step process, depicted in Figure 1:

1) Some subsets of data are selected. In Figure 1, ϕ selects possible subsets of data.
2) Each subset of data (both subset of columns and rows of the dataset) is assumed to be generalized by a single rule.

A subset of data is generalized by a rule. For a data subset $(\mathcal{A}, S)$ (some examples restricted to some attributes) of the dataset, the idea of the "generalization" function f from $(\mathcal{A}, S)$ is to generate the rule π. Thus, a data subset is generalized by a **single** rule.

This work investigates the f function, i.e., how to generate a rule from a subset of data. We do not tackle the question of determining ϕ, i.e., how to select data subsets. It is assumed that rules are generated from all possible selected subsets.

TABLE II
Ranges for the attributes of the dataset from Table I.

Attr.	*Range*	*Structure*
A_0	{cheap, low-priced, mid-priced, expensive}	total order
A_1	$[1, 500]$	metric
A_2	$\{1, 2, 3, 4, 5, 6\}$	metric
A_3	$\{A, B, C, D, E\}$	total order
A_4	{Toulouse, Rennes, Betton}	
A_5	{Downtown, Rangueil, Minimes}	
A_6	$\{S, N, W, E, SE, SW, NE, NW\}$	partial order

A. Data and rules

Data consist of tuples of size n (for n attributes $A_1, \cdots, A_n$). It is assumed that each tuple is assigned a class value. The range of an attribute A_i, denoted $\operatorname{Rng} A_i$, may, or may not, be ordered. For Table I, range of attributes and their structure are given in Table II. Values of attributes are of various types: floats (e.g., A_1), integers (A_2), discrete values either structured (e.g., A_6 can be partially ordered), or unstructured (e.g., A_4).

A rule learning algorithm elicits rules of the form:

$$A_{\pi(1)}(x) \in v_1 \wedge \cdots \wedge A_{\pi(k)}(x) \in v_k \rightarrow C(x) \in v_0 \qquad (*)$$

where $1 \leq k \leq n$, $v_i \subseteq \operatorname{Rng} A_{\pi(i)}$ for $i = 1..k$, $v_0 \subseteq \operatorname{Rng} C$ and $\{\pi(1), \cdots, \pi(k)\} \subseteq \{1, \cdots, n\}$.

Such a rule expresses that for an item x, if the value of each attribute $A_{\pi(i)}$ is one within v_i then the class value of x is one within v_0.

The value v_i is a subset of the range of the attribute $A_{\pi(i)}$ (or class C). So, v_i can be a singleton subset $\{u\}$ of the range $\operatorname{Rng} A_{\pi(i)}$ of the attribute i.e. $A_{\pi(i)}(x) \in v_i$ is $A_{\pi(i)}(x) = u$. Or, v_i can be a finite subset $\{u_1, \cdots, u_{i_p}\}$ of $\operatorname{Rng} A_{\pi(i)}$ hence $A_{\pi(i)}(x) \in v_i$ is just the disjunctive condition $A_{\pi(i)}(x) = u_1$ or $A_{\pi(i)}(x) = u_2$ or … or $A_{\pi(i)}(x) = u_{i_p}$. Disjunctive conclusions are unusual in rule learning but they could be desired and they generalize the approach. Lastly, v_i can be an arbitrary subset of the range $\operatorname{Rng} A_{\pi(i)}$. Structure over $\operatorname{Rng} A_{\pi(i)}$ can be exploited, e.g. $A_{\pi(i)}(x) \geq r$ is captured by setting v_i to the interval $[r, M]$ (if M is the greatest element).

B. General form of rules

We can identify a rule with a sequence of values for some attributes among $A_1, \cdots, A_n$ as well as C thus resulting in the general form for a rule π

$$\pi = A_{\pi(1)}(x) \in v_1^\pi \wedge \cdots \wedge A_{\pi(k_\pi)}(x) \in v_{k_\pi}^\pi \rightarrow C(x) \in v_0^\pi \quad (\dagger)$$

with $1 \leq k_\pi \leq n$, $v_i^\pi \subseteq \mathrm{Rng}\, A_{\pi(i)}$ for $i = 1..k_\pi$, $v_0^\pi \subseteq \mathrm{Rng}\, C$ and $\{\pi(1), \cdots, \pi(k_\pi)\} \subseteq \{1, \cdots, n\}$.

For the sake of simplicity, such a rule can be expressed as a member of $2^{\mathrm{Rng}\, C} \times 2^{\mathrm{Rng}\, A_1} \times \cdots \times 2^{\mathrm{Rng}\, A_n}$, i.e., a vector

$$\vec{\pi} = (v_0^\pi \ v_1^\pi \ v_2^\pi \cdots v_n^\pi) \qquad (\ddagger)$$

where for $i = 1..n$, $v_i^\pi = \mathrm{Rng}\, A_i$ if $A_i \notin \{A_{\pi(1)}, \cdots, A_{\pi(k_\pi)}\}$.

A tuple $(x_1, \cdots, x_n)$ which is assigned the class value c is said to be covered by the rule $\vec{\pi}$ above if $c \in v_0^\pi$ *and* all $x_i \in v_i^\pi$ (for $i \in \{\pi(1), \cdots, \pi(k_\pi)\}$).

Notation: In the sequel, S_i denotes the set of values that the attribute A_i takes in the subset S of the data, i.e.,

$$S_i \stackrel{\mathrm{def}}{=} \{x_i \mid (x_0, x_1, \cdots, x_n) \in S\}$$

C. Admissible rule generation

This work focuses on finding one rule, generalizing a subset of the dataset, which makes sense for the data scientist. Learning one rule aims at extending the set of values actually taken by the examples. Theoretically, any extra value, not covered by a counter-example, would work. However, some extended sets make more sense than others: we call this notion *rule admissibility*.

We do not consider how to find such a subset of the dataset but, given such a subset, we investigate the question of what is an admissible generalization of these examples for a user. Please note that we don't provide a definition for admissibility. What we provide is a framework to express different notions of "admissibility". Indeed, it depends upon application and users. The way we propose to analyze rule learning algorithms is to confront practical results with these different notions.

There is a sense in which a data subset determines a single rule. It is the view that the only rule generated by S is

$$A_1(x) \in \widehat{S_1} \wedge \cdots \wedge A_n(x) \in \widehat{S_n} \rightarrow C(x) \in \widehat{S_0} \qquad (\S)$$

where $\widehat{X}$ is the smallest rule admissible superset of X (for $X \subseteq \mathrm{Rng}\, A_i$ or $X \subseteq \mathrm{Rng}\, C$). Please note that this requires the assumption that attributes are independent for the purpose of rule admissibility.

That S is a data subset under ϕ (see Figure 1) means that if S is to amount to a rule π then each tuple in S is covered by π. Therefore, such a rule (§) is to be of the kind

$$A_{\pi(1)}(x) \in v_1 \wedge \cdots \wedge A_{\pi(k)}(x) \in v_k \rightarrow C(x) \in v_0 \qquad (**)$$

where $1 \leq k \leq n$, $S_{\pi(i)} \subseteq v_i$ for $i = 1, \cdots, k$ and $S_0 \subseteq v_0$ (as usual, $\{\pi(1), \cdots, \pi(k)\} \subseteq \{1, \cdots, n\}$).

What vector $(v_0 \ v_1 \ v_2 \cdots v_n)$ can count as a rule for the purpose of capturing S? Since $(**)$ is meant to capture S, we are looking for a vector $\vec{\pi} \geq (S_0 \ S_1 \ S_2 \cdots S_n)$ (i.e., $S_i \subseteq v_i^\pi$ for $i = 0, \cdots, n$) where every v_i^π is rule admissible.

Technically, the least[1] such π is the case that $v_i = S_i$ for $i = 0, \cdots, n$. As a rule, it does not fit. If S is to be viewed as a rule, the intuition is that a tuple close enough to some member(s) of S is expected to behave similarly to this member (or those members).

D. Properties of generalization as choice

The intuition we point out in the latter remark suggests that generalizing a set of values to a superset thereof amounts to applying a closure-like[2] operator $\widehat{\cdot}$. For any attribute A_i and data subset S, generalizing S_i is identified with mapping S_i to $\widehat{S_i}$, with properties taken from the list of Kuratowski's axioms:

$$\begin{array}{ll} \widehat{\emptyset} = \emptyset & \\ S \subseteq \widehat{S} \subseteq \mathrm{Rng}\, A_i & \\ \widehat{\widehat{S}} = \widehat{S} & \\ \widehat{S \cup S'} = \widehat{S} \cup \widehat{S'} & \text{(pre-closure)} \end{array}$$

Actually, we downgrade Kuratowski's axioms as follows (the Appendix reminds the definitions of related operators)

$$\begin{array}{ll} \widehat{S} \subseteq \widehat{S'} \text{ whenever } S \subseteq S' & \text{(closure)} \\ \widehat{S} = \widehat{S'} \text{ whenever } S \subseteq S' \subseteq \widehat{S} & \text{(cumulation)} \\ \widehat{S \cup S'} \subseteq \widehat{S} \text{ whenever } S' \subseteq \widehat{S} & \text{(capping)} \end{array}$$

We thus arrive at two classes of weaker operators that are worth exploring further: preclosure operators and capping operators, resp. realizing interpolation from single points and interpolation from pairs of points (with the view that rule generation encompasses interpolation of some kind).

This notion of rule admissibility is to be captured by a selection function, f, to fit the general view of Figure 1. Such a function (from a special class) determines an appropriate superset of S_i given some subsets of the powerset of $\mathrm{Rng}\, A_i$. The intuition here is that rule admissible subsets of the range $\mathrm{Rng}\, A_i$ of an attribute A_i can be characterized as *choices* from the powerset of $\mathrm{Rng}\, A_i$. Depending on what principles underly the actual choice, a different kind of closure embodies rule generation through the rule generalization principle.

The next theorems (with $\mathrm{Rng}\, A_i$ generalized to a set Z) specify two classes of selection functions that induce a closure-like operator over a powerset: preclosure and capping.

Theorem 1 (Selection functions inducing a preclosure operator): Let Z be a set such that $f : 2^{2^Z} \rightarrow 2^Z$ is a function satisfying the three conditions below for all $\mathcal{X} \subseteq 2^Z$ such that $\mathcal{X}$ is upward closed and all $\mathcal{Y} \subseteq 2^Z$:

1. $f(2^Z) = \emptyset$
2. $f(\mathcal{X}) \in \mathcal{X}$
3. $f(\mathcal{X} \cap \mathcal{Y}) = f(\mathcal{X}) \cup f(\mathcal{Y})$ whenever $\bigcup \min(\mathcal{X} \cap \mathcal{Y}) = \bigcup \min \mathcal{X} \cup \bigcup \min \mathcal{Y}$

The mapping $\widetilde{\cdot} : 2^Z \rightarrow 2^Z$ such that

$$\widetilde{X} \stackrel{\mathrm{def}}{=} f(\{Y \mid X \subseteq Y \subseteq Z\})$$

is a preclosure operator on Z.

[1] $\vec{\pi} \leq \vec{\pi'}$ iff $v_i^\pi \subseteq v_i^{\pi'}$ for $i = 0, \cdots, n$.

[2] Closure-like operators are topological operators, cf Appendix.

Theorem 2 (Selection functions inducing a capping operator): Let Z be a set, $f : 2^{2^Z} \to 2^Z$ be a function obeying the next two conditions for all $\mathcal{X} \subseteq 2^Z$ s.t. $\bigcap \mathcal{X} \in \mathcal{X}$ and all $\mathcal{Y} \subseteq 2^Z$:

1. $f(\mathcal{X}) \in \mathcal{X}$,
2. if $\mathcal{Y} \subseteq \mathcal{X}$ and $\exists H \in \mathcal{Y}$, $H \subseteq f(\mathcal{X})$ then $f(\mathcal{Y}) \subseteq f(\mathcal{X})$.

The mapping $\widetilde{\cdot} : 2^Z \to 2^Z$ such that

$$\widetilde{X} \stackrel{\text{def}}{=} f(\{Y \mid X \subseteq Y \subseteq Z\})$$

is a capping operator on Z.

III. Generation of rules with metric attributes

Back to the idea of rule generation as interpolation, we are considering the simplest case of rules with a collection of items that all take the value u for metric attribute A_i and that all are in class c. A rule for this case is

$$A_i(x) = u \to C(x) = c$$

i.e. $c = f_i(u)$ for some function f_i. This rule is restrictive as it only applies for items that take exactly the value u for A_i.

As items take a set of specific values $\{u_1, u_2, \ldots, u_n\}$, our idea is to generate rules that generalize them to an interval of values $[v, w]$. The main issue is to determine classes of selection functions that yield intuitive intervals of values.

The first approach is to propose a neighborhood principle. Since $c = f_i(u)$, it seems rather reasonable to still expect the class to be c for all values close enough to u. Assuming a notion of neighborhood, a rule exemplifying this would be

$$A_i(x) \in [u - r, u + r] \to C(x) = c.$$

This is developed in the next section where a class of selection functions is given that all induce a preclosure operator.

A drawback of the neighborhood approach is its predefined radius, r, which does not take into account the actual values distribution when it comes to finding intervals. Another section proposes a second approach that deals with interpolation from pairs (u, v) of values for an attribute A_i. This captures the idea that an interval of values is made of elements that are close enough to each other. We show that this principle can be captured through a capping operator.

A. Neighborhoods

As an application, consider neighborhoods for real-valued data (i.e., $\operatorname{Rng} A_i \subseteq \mathbb{R}$). For a datum $u \in \mathbb{R}$, we look at a generalization for u in the form of the neighborhood centered at u of radius r, for a given $r > 0$. For $r > 0 \in \mathbb{R}$, let $n_r : 2^{\mathbb{R}} \to 2^{\mathbb{R}}$ be the function:

$$n_r(X) \stackrel{\text{def}}{=} \begin{cases} \emptyset & \text{if} \quad X = \emptyset \\ \bigcup\limits_{u \in X} [u - r, u + r] & \text{otherwise} \end{cases}$$

It happens that n_r is a preclosure operator, i.e., as presented in the Appendix, n_r is a mapping $c : 2^{\mathcal{U}} \to 2^{\mathcal{U}}$ such that:

$c(\emptyset) = \emptyset$	(null fixpoint)
$X \subseteq c(X) \subseteq \mathcal{U}$	(extension)
$c(X \cup Y) = c(X) \cup c(Y)$	(preservation of binary unions)

Besides this independent evidence, we get the same conclusion from Theorem 1, giving an actual selection function f. Now, a useful abbreviation is $\uparrow\{X\} \stackrel{\text{def}}{=} \{Y \mid X \subseteq Y \subseteq Z\}$.

For a subset S of $\mathbb{R}$, define $f(\uparrow\{S\})$ by:

$$f(\uparrow\{S\}) \stackrel{\text{def}}{=} \begin{cases} \emptyset & \text{if} \quad S = \emptyset \\ \bigcup\limits_{x \in S} [x - r, x + r] & \text{otherwise} \end{cases}$$

Extend f to all of 2^{2^Z} by taking

$$f(\mathcal{X}) \stackrel{\text{def}}{=} \bigcup_{S \in \min \mathcal{X}} f(\uparrow\{S\}).$$

(Since $\mathcal{X}$ denotes a collection of subsets of Z, $\min \mathcal{X}$ denotes those sets in $\mathcal{X}$ that have no proper subset in $\mathcal{X}$.)

Then, f satisfies conditions 1-3 of Theorem 1.

B. Intervals

We look now at generalization from S_i as interpolation of the kind: If $u \in S_i$ and $v \in S_i$ such that the distance between u and v is smaller than some threshold then generalize u and v to all values (in the range of A_i) between u and v. We again follow the idea that generalizing S_i amounts to applying some kind of closure operator $\widehat{\cdot}$, giving $\widehat{S_i}$. We start with considering closure operators (see the Appendix), i.e., for all $S \subseteq \operatorname{Rng} A_i$ and $S' \subseteq \operatorname{Rng} A_i$, the following holds:

$$\begin{aligned} &S \subseteq \widehat{S} \subseteq \operatorname{Rng} A_i, \\ &\widehat{\widehat{S}} = \widehat{S}, \\ &\widehat{S} \subseteq \widehat{S'} \text{ whenever } S \subseteq S'. \end{aligned}$$

Interestingly, for a data subset S, in order to determine *the* rule π induced by S, applying $\widehat{\widehat{S}} = \widehat{S}$ means that if S_i is rule admissible then it is enough to set $v_i^\pi = S_i$ and no further adjustment over π is needed regarding the attribute A_i (further adjustements are likely for other attributes).

C. Example of $\widehat{\cdot}$ not being a closure operator

Imagine a principle that generalizes values (from $\mathbb{N}$) to small intervals over $\mathbb{N}$. For instance, let attribute A_2 be *distance to townhall*. Let the class attribute C be *level of rent* (understood as ranging over cheap, low-priced, ...). From the minimalistic S consisting of items 1 and 2 below:

	A_1	*Distance to townhall*	...	*Level of rent*
item 1	...	3 km	...	cheap
item 2	...	7 km	...	cheap

then such a principle could make $S = \{\text{item}\,1, \text{item}\,2\}$ to generate a rule π with $v_2^\pi = [3, 7]$ (i.e., 3 km to 7 km). Intuitively, the rule would express that flats for rent within 3 to 7 km from the town hall are most affordable. However, from S' consisting of items 1 to 4 as follows:

	A_1	*Distance to townhall*	...	*Level of rent*
item i'_1	...	3 km	...	cheap
item i'_2	...	7 km	...	cheap
item i'_3	...	1 km	...	cheap
item i'_4	...	9 km	...	cheap

the same principle can make $S' = \{i'_1, \ldots, i'_4\}$ to give a rule with $v_2^{\pi'} = [1,3] \cup [7,9]$. This is a counterexample to closure because (isotony) fails: $S_2 \subseteq S'_2$ but $v_2^{\pi} \not\subseteq v_2^{\pi'}$.

π' says that rents of flats in the vicinity of the town hall are low and so are rents of flats in a 7 to 9 km ring from the town hall but no example confirm this for flats in the range 3 to 7 km. We can regard [1,3] ∪ [7,9] as more admissible than the large [1,9].

D. Example of $\widehat{\cdot}$ not being a cumulation operator

Since $\widehat{\cdot}$ fails to be a closure operator, the next possibility is that $\widehat{\cdot}$ is a cumulation operator (every closure operator is a cumulation operator but the converse is untrue).

Again, think of some principle that generalizes values (from $\mathbb{N}$) to small intervals over $\mathbb{N}$. Here is a brief example (we use decimals of km to abbreviate hundreds of meters), with S consisting of items i_1 to i_9 below:

	A_1	*Distance to townhall*	…	*Level of rent*
item 1	…	4.1 km	…	cheap
item 2	…	4.4 km	…	cheap
item 3	…	4.8 km	…	cheap
item 4	…	5 km	…	cheap
item 5	…	5.5 km	…	cheap
item 6	…	5.8 km	…	cheap
item 7	…	6.1 km	…	cheap
item 8	…	6.6 km	…	cheap
item 9	…	6.9 km	…	cheap

Then, such a principle could make S (the above 9 items) to generate a rule with $v_2^{\pi} = [4.1, 6.9]$. Indeed, gaps between any two consecutive values among these nine values are of somewhat similar length and are turned into intervals. Now, from S' consisting of items i'_1 to i'_{20} below:

	A_1	*Distance to townhall*	…	*Level of rent*
item i'_1	…	4.1 km	…	cheap
item i'_2	…	4.2 km	…	cheap
item i'_3	…	4.3 km	…	cheap
item i'_4	…	4.4 km	…	cheap
item i'_5	…	4.6 km	…	cheap
item i'_6	…	4.7 km	…	cheap
item i'_7	…	4.8 km	…	cheap
item i'_8	…	6.6 km	…	cheap
item i'_9	…	6.9 km	…	cheap
item i'_{10}	…	4.1 km	…	cheap
item i'_{11}	…	4.1 km	…	cheap
item i'_{12}	…	4.4 km	…	cheap
item i'_{13}	…	4.8 km	…	cheap
item i'_{14}	…	5 km	…	cheap
item i'_{15}	…	5.5 km	…	cheap
item i'_{16}	…	5.8 km	…	cheap
item i'_{17}	…	6.1 km	…	cheap
item i'_{18}	…	6.6 km	…	cheap
item i'_{19}	…	6.8 km	…	cheap
item i'_{20}	…	6.9 km	…	cheap

then the same principle can make S' (items i'_1 to i'_{20}) to give a rule with $v_2^{\pi'} = [4.1, 5] \cup [5.5, 6.9]$. Again, the idea is that the gap between two consecutive values is to be turned into an interval unless the gap is much greater than most of the other gaps in the series: the gap from 5.0 to 5.5 has length .5 but all other gaps here (from 4.1 to 4.2, …, from 6.8 to 6.9) have length at most .2.

It is a counterexample to cumulation because $S_2 \subseteq S'_2 \subseteq v_2^{\pi}$ but $v_2^{\pi} \not\subseteq v_2^{\pi'}$ (still, $v_2^{\pi'} \subseteq v_2^{\pi}$).

All this suggests some kind of preservation principle:

> *If new items confirming a rule are added, generalization should not make the rule to be further generalized.*

It seems that such an approach to generalizing a set of values to a superset thereof amounts to applying a capping operator. The next section shows that such a view can be identified with using a selection function (from the special class specified in Theorem 2) to determine the appropriate superset eliciting the rule.

E. Capping operator: selection via power means

Let the special case that Z is totally ordered and f selects, among all supersets of $S = \langle x_1, \ldots, x_m \rangle$,[3] the union of the intervals over Z that have both endpoints in S and length (denoted by l) bounded by a threshold value $\Delta(S)$ as follows

$$\varphi_S(x_j) \stackrel{\text{def}}{=} \begin{cases} [x_j, x_{j+1}] \text{ if } l([x_j, x_{j+1}]) \leq \Delta(S) \\ [x_j, x_j] \quad \text{else (including } j = m) \end{cases} \tag{1}$$

where Δ is a function on the increasing sequences over Z, i.e. $S \mapsto \Delta(S)$ for every increasing sequence S.

For all finite $S \subseteq Z$, define

$$f(\uparrow\{S\}) \stackrel{\text{def}}{=} \bigcup_{x \in S} \varphi_S(x) \tag{2}$$

where the $\varphi_S : S \to 2^Z$ functions can be required to satisfy, for all $x \in S$ and all finite $S' \subseteq Z$, the following constraints

(i) $\quad x \in \varphi_S(x)$,

(ii) $\quad S \subseteq S' \subseteq \bigcup \varphi_S(S) \;\Rightarrow\; \bigcup \varphi_{S'}(S') \subseteq \bigcup \varphi_S(S)$.

Extend f to all of 2^{2^Z} by taking

$$f(\mathcal{X}) \stackrel{\text{def}}{=} \bigcap \mathcal{X} \text{ whenever } \mathcal{X} \neq \uparrow\{S\} \text{ for all } S \subseteq Z. \tag{3}$$

Proposition 1: If φ satisfies conditions (i) and (ii) then f as defined by *(2)-(3)* enjoys conditions 1.-2. of Theorem 2.

We focus on intervals with endpoints in $\mathbb{R}$ (hence $S \subseteq \mathbb{R}$) and length the absolute difference between both endpoints.

Proposition 2: Let φ be as in *(1)* with Δ such that for all finite S and S', if $S \subseteq S' \subseteq \bigcup \varphi_S(S)$ then $\Delta(S') \leq \Delta(S)$. If Δ is real-valued, if Z is $\mathbb{R}$ and if $l([x, y]) = |y - x|$ then φ satisfies (i)-(ii).

An almost direct application of Theorem 2 then implies that functions defined as in (1) induce capping operators. The

[3] Here, S is identified with its enumeration in increasing order since this simplifies the formulation in (1).

following instances for Δ functions give selection functions, as per (1)–(3), generating admissible rules:

- Geometric mean: $\Delta(S) = \left(\prod_{i=1}^{m-1}(x_{i+1} - x_i)\right)^{\frac{1}{m-1}}$
- Higher power means: $\Delta(S) = \sqrt[p]{\frac{1}{m-1}\sum_{i=2}^{m}(x_i - x_{i-1})^p}$

IV. Related work

Rule learning algorithms are a class of machine learning algorithms mainly developed in the 90's [22] that drew recent interest due to the interpretable nature of its outputs [8].

Prior works have proposed foundations for rule learning and many algorithms exist. A major reference is [8] that broadly presents the concepts used in rule learning algorithms. It mainly focuses on practical aspects that enable the reader to understand a broad range of algorithms. Our framework aims at turning the rule learning formalization to a more conceptual level. We investigate in particular the LearnOneRule step of the rule learning process. A key issue in the LearnOneRule algorithm is how to evaluate and compare different rules [8] using several measures such as precision, information gain, correlation, m-estimate, etc. The basic principle underlying these measures is a simultaneous optimization of consistency and coverage. This optimization addresses the way of choosing a subset of examples covered by a rule but does not give any information on the interest of a generalization from a data scientist viewpoint. Our framework allows to address admissibility of generalizations and thus to define classes of generalizations able to catch empirical concepts of neighborhood for example.

This notion of admissibility contributes to a formalization of the interpretativeness of the outputs of rule learning algorithms. Similar questions have been addressed in previous works. To the best of our knowledge, none of them addressed the problem of the choice of the values in the rules. They take into consideration the structure of the rules (e.g. their size) or the rule set [4], [2], [21], [17], [3]. Instances of the GUHA method to mining association rule [12] fall under our approach if conclusions of such rules are to play the role of classes.

In [1], the proposed framework is based on a score for rule quality measures. It does not use quality measures that select intuitive rules, but the most accurate ones. [15] addressed the intuitiveness of rules through the effects of cognitive biases. They notice that a number of biases can be triggered by the lack of understanding of attributes or their values appearing in rules. In [9], the authors suggest that "longer explanations may be more convincing than shorter ones" (see [20], too) and evaluate this criterion using a crowd-sourcing study based on about 3.000 judgments.

Hence, one of our contributions is to relate generalization of examples to closure-like operators. Relationship with Formal Concept Analysis [10] then comes to mind. In [14], the authors investigate the problem of mining numerical data with Formal Concept Analysis. This amounts to a way to generate some subsets of data. As we have shown that the operators at work in generalizing by intervals are weaker than closure operators, no equivalence is expected. Even the idea that a subset $(\mathcal{A}, S)$ of the dataset is always a subset of a concept fails in general. For instance, let ϕ capture the idea of "contraries", in which case a selected subset of size two consists in two examples e_1 and e_2 such that $A_i(e_1) \neq A_i(e_2)$ for all A_i in $\mathcal{A}$ hence $\sigma(\{e_1, e_2\}) = \emptyset$ which entails that no superset of $\{e_1, e_2\}$ can be a concept with a non-empty set of attributes. In contrast, there exist selection functions that provide a subset of $\operatorname{Rng} A_i$ as a generalization for $\{e_1, e_2\}$.

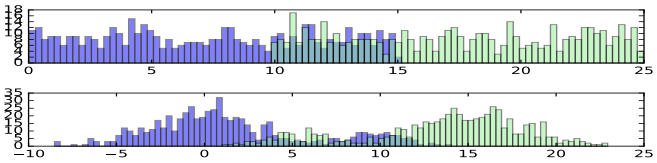

Fig. 2. Data distributions for classes `blue` (in blue) and `green` (in green). In the upper histogram, the data distributions are simulated using uniform distributions. In the lower histogram, the data distributions are simulated using a mixture of two normal distributions per class.

V. Illustration with CN2 rules

We illustrate some behaviours of CN2 [5] when facing artificial data distributions, to show that our proposed formalisation offers a framework for analyzing rule learning algorithms. We use a simulated dataset with a single numerical attribute, v, and two classes. The form of the generated rules is $v \in [l, u] \Rightarrow C$ where $[l, u]$ is an interval and $C \in \{\texttt{blue}, \texttt{green}\}$.

Remember, the abstract modeling of a rule learning process has two steps: selection of a subset of data and generalisation of this subset of data by a rule. This article is focused on the generalisation step. The simple case studies below make the assumption that each class of the dataset corresponds to a subset of examples to generalize. Then, our analysis assumes two subsets of data (and thus two rules): the subset of data labeled as `blue` and the ones labeled as `green`. These two subsets are for illustration purposes only, we make no claim that they are more sensible than other alternative subsets.

A. CN2 splits potentially interesting intervals

Here, we illustrate the fact that, despite the relative continuity of the attribute, the CN2 algorithm splits attribute intervals of a rule in some specific cases of overlapping values.

Figure 2 illustrates two data distributions for which we run the CN2 algorithm. The distribution of each class is dense from lower to higher value, i.e., the gaps between two consecutive examples of a class are bounded. It is desirable to have exactly one rule per class generalizing all examples. This would be the case if rule generalization were to follow the principles of neighborhoods or intervals applied on data subsets made of examples belonging to the same class. Yet, the CN2 algorithm splits the interval in several sub-intervals to improve its selection criteria based on accuracy.

The extracted rules in case of uniform distributions (top of Figure 2), resp., for normal distributions (bottom of Figure 2)

are in the leftmost list, resp., in the rightmost list below:

- $v \in [-\infty, 10.03] \Rightarrow$ `blue`
- $v \in [12.73, 14.83] \Rightarrow$ `blue`
- $v \in [10.65, 12.81] \Rightarrow$ `green`
- $v \in [15.01, \infty] \Rightarrow$ `green`

- $v \in [-\infty, 0.96] \Rightarrow$ `blue`
- $v \in [0.97, 2.57] \Rightarrow$ `blue`
- $v \in [3.09, 10.04] \Rightarrow$ `blue`
- $v \in [3.50, 7.18] \Rightarrow$ `green`
- $v \in [11.55, 13.14] \Rightarrow$ `green`
- $v \in [13.15, \infty] \Rightarrow$ `green`

In the case of the uniform distributions, the intervals of rules with different decision classes may overlap. CN2 thus allows for intervals occurring in different rules to overlap.

B. Impact of example density on boundaries choice

Figure 3 illustrates the case of two (single-attribute) datasets whose class distributions are similar: uniform distribution with the same bounds, but different intensities. Example-classes are balanced in the first dataset but not in the second.

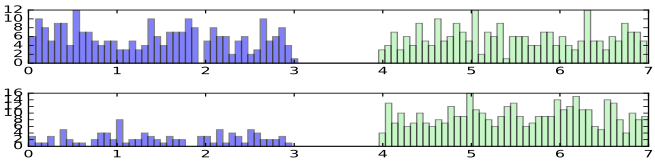

Fig. 3. Balanced (on top) and unbalanced (on bottom) distributions of two classes examples. Classes are separated by a fixed distance.

The idea here is to study the impact of multiple instances on the choice of boundaries by CN2. As per the function inducing a capping operator, Δ has to be non-decreasing over confirming examples. So, having more examples leads to change the boundaries of the rule that generalizes examples. But in both cases, CN2 finds the very same bound 3.49: $v \in [-\infty, 3.49] \Rightarrow$ `blue` and $v \in [3.49, \infty] \Rightarrow$ `green`.

This illustrates the more general situation that an approach insensitive to density of examples over the choice of boundaries amounts to a cumulation operator:

Theorem 3: Let Z be a set such that $f : 2^{2^Z} \to 2^Z$ is a function satisfying the two conditions below for all nonempty $\mathcal{X} \subseteq 2^Z$ and $\mathcal{Y} \subseteq 2^Z$:

1. $f(\mathcal{X}) \in \mathcal{X}$,
2. $f(\mathcal{X} \cup \mathcal{Y}) \in \mathcal{X} \;\Rightarrow\; f(\mathcal{X} \cup \mathcal{Y}) = f(\mathcal{X})$.

The mapping $\overline{\cdot} : 2^Z \to 2^Z$ such that

$$\overline{X} \stackrel{\text{def}}{=} f(\{Y \mid X \subseteq Y \subseteq Z\})$$

is a cumulation operator on Z.

C. Impact of example sparsity on boundaries choice

Figure 4 illustrates two datasets whose example distributions differ by the gaps between consecutive examples of the same class. In the first case, these gaps are small (average gap of 1 unit) w.r.t. the gap between the two classes (3 units). In such a case, we expect to generate a rule that gather all examples in a single interval. It is actually what happens with CN2. In the second case, gaps between consecutive examples are larger (average gap of 10 units), but the behaviour of CN2 remains

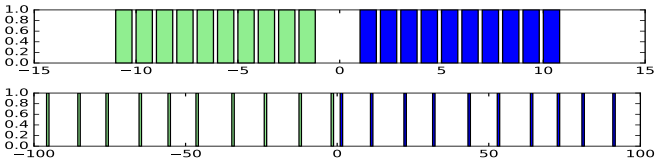

Fig. 4. Two datasets centered on 0 (each bar is one example). The gap between consecutive examples is 1 unit at the top and 10 units at the bottom.

the same. Our experiments show that the gap length has no consequences on the rules. The very same rule is generated splitting examples by comparing their value with 0. We can conclude from this example that CN2 does not behave in the way described by preclosure operators.

The two preceding experiments seem to show that rules are generated only from the extreme values of examples sets without considering the actual example distribution. Such a behaviour appears to be more constrained than the one of a capping operator. It amounts to a Δ function that would not be decreasing. But these examples are specific cases of datasets with well-separated classes. In case of overlapping range of values (see Figure 2), the rule choices that have been made depend on example distributions.

VI. Conclusion and perspectives

Evaluation and comparison of rules in the rule learning task are currently based on optimization of consistency and coverage measures. In this article, we look at the notion of rule admissibility as the interest of a generalization from a data scientist viewpoint. We define a framework providing a formal approach to the generalization of examples in the attribute-value rule learning task and some foundations for the admissible generalizations of examples as rules. Our notion of admissibility is presented as a choice of one generalization among all supersets of examples. Distinguished notions of choice are shown to capture a closure-like operator (preclosure or capping). In the case of metric attributes, we offer actual selection functions that induce such operators.

These selection functions show how our framework supports the analysis of rule-learning evaluation. We have generated synthetic datasets to analyze the behaviour of CN2 in view of notions arising from our framework. Thus, we point out some counter-intuitive behaviours of this algorithm.

Some novelty in our work lies with it focussing on the values occuring in rules, instead of, e.g., the structure of rules.

Since rule learning may involve non-numerical attributes, a short term perspective is to also give selection functions for attributes with weaker structure than enjoyed by numerical attributes (metric structure). Finally, this article does not address the issue of selection of data subsets. Future work is to focus on this part of the process to propose a more complete formal model for attribute-value rule learning.

Appendix

Given a set $\mathcal{U}$, a *Kuratowki closure operator* [16] is a mapping $c : 2^{\mathcal{U}} \to 2^{\mathcal{U}}$ such that for all $X \subseteq 2^{\mathcal{U}}$ and all $Y \subseteq 2^{\mathcal{U}}$

$c(\emptyset) = \emptyset$ (null fixpoint)
$X \subseteq c(X) \subseteq \mathcal{U}$ (extension)
$c(X) = c(c(X))$ (idempotence)
$c(X \cup Y) = c(X) \cup c(Y)$ (preservation of binary unions)

A first direction to weaken Kuratowki closure operators is to drop (idempotence), resulting in *preclosure operators*

$c(\emptyset) = \emptyset$ (null fixpoint)
$X \subseteq c(X) \subseteq \mathcal{U}$ (extension)
$c(X \cup Y) = c(X) \cup c(Y)$ (preservation of binary unions)

Another direction amounts to dropping (null fixpoint) and replacing (preservation of binary unions) by a weaker axiom with all this giving *abstract closure operators*

$X \subseteq c(X) \subseteq \mathcal{U}$ (extension)
$c(X) = c(c(X))$ (idempotence)
$X \subseteq Y \;\Rightarrow\; c(X) \subseteq c(Y)$ (isotony)

These, in turn, can be weakened (various axioms weaker than (isotony) are detailed in [18]) to *cumulation operators*

$X \subseteq c(X) \subseteq \mathcal{U}$ (extension)
$X \subseteq Y \subseteq c(X) \;\Rightarrow\; c(X) = c(Y)$ (cumulation)

which can themselves be weakened to *capping operators*[4]

$X \subseteq c(X) \subseteq \mathcal{U}$ (extension)
$Y \subseteq c(X) \;\Rightarrow\; c(X \cup Y) \subseteq c(X)$ (capping)

For cumulation and capping operators, (idempotence) holds as it is actually a consequence of the other two axioms.

Sketch of proofs

Proof: [Theorem 1] (Preservation of binary unions) For $\mathcal{X} = \uparrow\{X\}$ and $\mathcal{Y} = \uparrow\{Y\}$, the proviso for condition 3. is $\bigcup\min(\uparrow\{X\} \cap \uparrow\{Y\}) = \bigcup\min \uparrow\{X\} \cup \bigcup\min \uparrow\{Y\}$. However, $\uparrow\{X\} \cap \uparrow\{Y\} = \uparrow\{X \cup Y\}$ hence the proviso becomes $\bigcup\min(\uparrow\{X \cup Y\}) = \bigcup\min \uparrow\{X\} \cup \bigcup\min \uparrow\{Y\}$ i.e. $X \cup Y = X \cup Y$ (as $\min(\uparrow\{W\} = \{W\}$). Condition 3. gives $f(\uparrow\{X\} \cap \uparrow\{Y\}) = f(\uparrow\{X\}) \cup f(\uparrow\{Y\})$. Applying $\uparrow\{X\} \cap \uparrow\{Y\} = \uparrow\{X \cup Y\}$ once again, $f(\uparrow\{X \cup Y\}) = f(\uparrow\{X\}) \cup f(\uparrow\{Y\})$. Equivalently, $\widetilde{X \cup Y} = \widetilde{X} \cup \widetilde{Y}$. ∎

Proof: [Theorem 2] (Capping) Assume $Y \subseteq \widetilde{X}$ i.e. $Y \subseteq f(\uparrow\{X\})$. 1. can be applied to give $X \subseteq f(\uparrow\{X\})$. Therefore, $X \cup Y \subseteq f(\uparrow\{X\})$. In view of $X \cup Y \in \uparrow\{X \cup Y\}$, this gives $\exists H \in \uparrow\{X \cup Y\}$ such that $H \subseteq f(\uparrow\{X\})$. Also, $\uparrow\{X \cup Y\} \subseteq \uparrow\{X\}$ because $\uparrow$ is antitone. Applying now 2., $f(\uparrow\{X \cup Y\}) \subseteq f(\uparrow\{X\})$ ensues, i.e., $\widetilde{X \cup Y} \subseteq \widetilde{X}$. ∎

References

[1] José L. Balcázar and Francis Dogbey. Evaluation of association rule quality measures through feature extraction. In Allan Tucker, Frank Höppner, Arno Siebes, and Stephen Swift, editors, *Proceedings of the 12th International Symposium on Advances in Intelligent Data Analysis XII (IDA'2013)*, volume 8207 of *Information Systems and Applications*, pages 68–79, London, UK, October 2013. Springer.

[2] Fernando Benites and Elena Sapozhnikova. Hierarchical interestingness measures for association rules with generalization on both antecedent and consequent sides. *Pattern Recognition Letters*, 65:197–203, 2015.

[3] Mario Boley, Bryan R. Goldsmith, Luca M. Ghiringhelli, and Jilles Vreeken. Identifying consistent statements about numerical data with dispersion-corrected subgroup discovery. *Data Mining and Knowledge Discovery*, 31(5):1391–1418, September 2017.

[4] Alberto Cano, Amelia Zafra, and Sebastián Ventura. An interpretable classification rule mining algorithm. *Information Sciences*, 240:1–20, 2013.

[5] Peter Clark and Tim Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.

[6] William W Cohen. Fast effective rule induction. In Armand Prieditis and Stuart J. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning (ICML'1995)*, pages 115–123, Tahoe City, CA, USA, July 1995. Morgan Kaufmann.

[7] Hugo Jair Escalante, Sergio Escalera, Isabelle Guyon, Xavier Baró, Yağmur Güçlütürk, Umut Güçlü, and Marcel A. J. van Gerven, editors. *Explainable and Interpretable Models in Computer Vision and Machine Learning*. The Springer Series on Challenges in Machine Learning. Springer, 2018.

[8] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of Rule Learning*. Springer Science & Business Media, 2012.

[9] Johannes Fürnkranz, Tomás Kliegr, and Heiko Paulheim. On cognitive preferences and the plausability of rule-based models. *CoRR*, abs/1803.01316, 2019.

[10] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer Science & Business Media, 2012.

[11] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys*, 38(3), 2006.

[12] Petr Hájek, Martin Holena, and Jan Rauch. The GUHA method and its meaning for data mining. *Computer System Science*, 76(1):34–48, 2010.

[13] Jon Hills, Anthony J. Bagnall, Beatriz de la Iglesia, and Graeme Richards. BruteSuppression: a size reduction method for apriori rule sets. *Intelligent Information Systems*, 40(3):431–454, 2013.

[14] Mehdi Kaytoue, Sergei O. Kuznetsov, and Amedeo Napoli. Revisiting numerical pattern mining with formal concept analysis. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'2011)*, pages 1342–1347, Barcelona, Catalonia, Spain, July 2011. IJCAI/AAAI Press.

[15] Tomás Kliegr, Stepán Bahník, and Johannes Fürnkranz. A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. *CoRR*, abs/1804.02969, 2018.

[16] Kazimierz Kuratowski. *Topology*, volume I. Academic Press, 1966.

[17] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2016)*, pages 1675–1684, San Francisco, CA, USA, August 2016. ACM.

[18] David Makinson. General patterns in nonmonotonic reasoning. In Dov M. Gabbay, Chris J. Hogger, and John Alan Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume III (Donald Nute, volume co-ordinator). Clarendon Press, Oxford, 1994.

[19] Tom M Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[20] Julius Stecher, Frederik Janssen, and Johannes Fürnkranz. Shorter rules are better, aren't they? In Toon Calders, Michelangelo Ceci, and Donato Malerba, editors, *Proceedings of the 19th International Conference on Discovery Science (DS'2016)*, volume 9956 of *Lecture Notes in Computer Science*, pages 279–294, Bari, Italy, October 2016. Springer.

[21] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. Bayesian rule sets for interpretable classification. In Francesco Bonchi, Josep Domingo-Ferrer, Ricardo A. Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu, editors, *16th IEEE International Conference on Data Mining (ICDM'2016)*, pages 1269–1274, Barcelona, Spain, December 2016. IEEE.

[22] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 4th edition, 2016.

[4] Capping originates in logic where it is called Restricted Cut as it captures the principle that intermediate conclusions can be freely removed from the premises with no loss among conclusions.